LA-UR-04-2149

Approved for public release; distribution is unlimited.

Title: Representing Subjective Knowledge in Engineering Systems using Possibility Trees

Author(s): Sunil Donald, Terry F. Bott, and Stephen W. Eisenhawer
Los Alamos National Laboratory

Submitted to: 8th World Multi-Conference on Systemics, Cybernetics and Informatics
Orlando, Florida
July 18–21, 2004

Representing Subjective Knowledge in Engineering Systems Using Possibility Trees

Sunil Donald, Terry F. Bott, and Stephen W. Eisenhawer
Los Alamos National Laboratory
Los Alamos, NM 87545, USA

ABSTRACT

Representation of knowledge in a complex system can be a challenging task, particularly when the structure of the system is novel or evolving. In such instances, understanding the system may be mostly subjective. Subjective system knowledge can be very useful in analyzing system behavior if a suitable means is developed to extract what information does exist and represent it coherently. As the novelty and complexity of the system increases, the need to organize and represent subjective knowledge increases rapidly. To be maximally useful, subjective knowledge should be represented using a structure that (1) is sufficiently flexible to accommodate the various forms of information that may be encountered, (2) provides a logical organization of the information, and (3) adheres to a set of formal rules. A logic-gate tree provides such a structure—large amounts of information from diverse sources can be organized in a structured form with complex relationships between pieces of information (hereafter called "elements") represented by logic gates. Using this structure, the collected information can be combined algorithmically to generate knowledge about complex aspects of the behavior of the system that would be difficult to divine using less-structured approaches. This paper presents the theory and underlying motivations of a possibility tree, along with a few representative examples to illustrate the application of the tree to real-world systems.

Keywords: Subjective knowledge, possibility trees, logic gates, scenarios.

1. INTRODUCTION

Two types of trees, fault trees and event trees, have been used successfully to organize knowledge about the failure processes of well-characterized engineering systems [1]. Such trees are applied most successfully to systems with relatively well-defined components, subsystems, and procedural actions and where faults cause specified system effects. A system is defined as any collection of interrelated and interdependent objects, both human and mechanical components, that contributes to the overall behavior of the system. A system's behavior is dictated by its components as well as by the threats it is exposed to, and experts analyze the system according to the required information. Usually when considering engineering systems, experts rely on knowledge about components of the system under consideration and use the trees to organize the interdependent components. For example, in the case of a nuclear power plant, wherein the threat is from a malfunction of a component, system behavior can be predicted by organizing all the elements that contribute to the malfunction and then determining the effect of this malfunction on the overall behavior of the system using a fault tree and an event tree. Such a representation has been applied successfully in many analyses and continues to be effective in the risk analysis of high-risk systems such as power plants. However, when the system's behavior is dependent on components that are not well understood and the system's exposure is a function of internal or external threats that are only vaguely understood, an analysis of the system cannot be accomplished using the restrictive mechanism offered by fault trees and event trees. Thus, in analyzing systems that are only vaguely understood, an extension of the ideas underlying fault trees and event trees is required to capture those portions of the system that are partially known. When considering such systems, the components of the system and their response to the threat are known only through subjective assessments of subject-matter experts (SMEs). This subjective assessment usually is based on experts' experience with that particular system and their perception of events that may or may not occur in the future. In such assessments, expert information is crucial to analyzing the system. However, because experts are limited by their cognitive ability to analyze all of the information relevant to the topic, the process of extracting information from the experts can be fraught with gaps and lead to an incomplete description of the system. As the complexity of the system increases, there are more opportunities for experts to overlook the details and thereby affect the analysis in a significant manner. This paper discusses a logic-gate tree called a "possibility tree" that can be used to extract information from experts systematically in a compact tree form so that the information is manageable and accessible for further analysis. Possibility trees already have been used successfully in a few engineering applications [2, 3].

2. POSSIBILITY TREE

The main goal of a possibility tree is to organize information and allow an analyst to study the behavior of complex systems using a tree structure. The result of the application of a possibility tree is a systematic, consistent, and logical evaluation of all available information used to generate possible risk scenarios and subsequently lead to an analysis of those scenarios. This model offers two important benefits in analyzing uncertainty: (1) it allows an efficient organization of expert knowledge of the system in a tree form such that information is traceable and easily accessible, and (2) it enables the analyst to make inferences even when data are missing. The latter benefit is enabled by the use of approximate reasoning methods that can be useful in analyzing the risks [4].

In a possibility tree, the fault-tree concept is extended by expanding the logic-gate set used in fault-tree construction and by removing the standard assumptions of inevitable causality and commutation of elements completely. The logical purpose served by event trees is incorporated in the possibility-tree structure by allowing development of interlinked logic-gate trees from the bottom up (leaves to branches) as well as from the top down (branches to leaves). Using a possibility tree, a set of experts and analysts can generate useful models of behavior for even vaguely defined systems. Such models are based on partial knowledge about the system and are developed by deductively generating all possible event structures through a systematic exercise of their knowledge at each node and level of the tree. The fundamental assumption underlying the use of the possibility tree is that complex system behavior can be modeled by logically connecting sets of discrete events and states, called the "elements" of the tree. This fundamental assumption is rendered less restrictive by introducing logic gates that model complex relationships between elements such as cycles and conditional branching. By means of these extensions, the possibility tree becomes a way to develop generalized graphs deductively by using the logic gate as a shorthand way of expressing relationships between nodes.

A possibility tree, more formally, is defined as a data structure T = {G, E}, where G is the set of logic gates that encode aggregation logic and E defines the edges that link information pieces. The set of logic gates G is given as G = {Terminal, And, Or, Exclusive Or, Causal, Taxonomy, Cycle} and determines how the information pieces are linked. A possibility tree is more appropriately viewed as a logic-gate tree and differs from a regular tree in that the structure is more general and can include cycles. Thus, information can be represented with fewer constraints on what information can be included. Additionally, the set of gates used in the tree can be expanded to include new and novel processes.

To construct a possibility tree, information about system processes is extracted from sources of general knowledge, expert judgments, and observations, by analogy, or through heuristics. This knowledge is converted into discrete elements that are linked together using logic gates as connectives. The system characteristics thus are uncovered deductively from all known sources of relevant information using step-by-step causality-based reasoning. This reasoning process produces a hierarchical tree structure with well-defined connections between levels of the tree. The tree structure often can be used to capture competing views about the possible causes for various events in a single-tree structure.

The knowledge elements are linguistic descriptions of pieces of knowledge. The logical connectives, linguistic in nature themselves, determine the structure of the logical equation underlying each possible path in the tree and characterize the behavior of the system. The paths are generated from the representative elements by evaluating the logical equation from a leaf to the root of the tree according to the type of connective. In essence, each of these generated paths represents a possible process written out in natural language that can lead to a given system state. Through the generation of paths, a possibility tree allows experts to analyze system behavior even in the absence of quantitative data. A possibility tree thus offers a convenient, intuitive, and logical way to extract system characteristics from subjective information in complex systems analysis.

3. AGGREGATION

All nodes connected to a particular gate are aggregated according to the logic encoded in that gate. In contrast to fault trees and event trees where the logic of the gate defines the Boolean aggregation of the input nodes, possibility-tree logic defines the nature of aggregating natural language statements. Thus, the gates determine the logical relationship of the input nodes. Logic gates are defined by the following operations.

Exclusive Or:
An "Exclusive Or" (EOR) gate describes an operation where only one of the inputs is required to be sufficiently possible for the path to exist. Therefore, paths are formed by selecting each element from the set of inputs.

And:
An "And" gate defines a path that is a combination of the inputs into the gate and consists of concatenating all possible inputs into the gate.

Or:
An inclusive "Or" gate determines paths such that any or all of the combinations of the inputs into the gate can be possible.

Causal:
A "Causal" gate is an ordered gate, where the inputs are ordered according to the sequence of events that are to be realized for the path to exist and the events are arranged in a chain depicting causality.

Cycle:
A "Cycle" gate, commonly used to define processes that consist of cyclical behavior, consists of inputs that are repeated until a specified condition is satisfied. The cycle then exits to a node in the tree that is identified by the condition.

Taxonomy:
Inputs into a "Taxonomy" gate usually are used to classify information relevant to the gate. The aggregation is similar to an EOR gate, with each input into the gate taken individually to form a path.

The gates are used according to their relevance to the detail at a particular level of the tree, and they can be chained together according to the required logical connection. For example, when an EOR gate forms one of the inputs into an "And" gate, the combination of the inputs into the gate is described by a Cartesian product of all elements of an EOR gate, with other inputs into the "And" gate. Thus, the number of paths that are generated from the tree usually depends on the amount of required detail represented in the tree and is proportional to the number of nodes and levels in the tree and the types of logical gates used to connect the nodes. In general, the following equation defines the number of paths generated at each gate of the tree:

$$N_p = \begin{cases} 1 & \text{Node Type} = \text{Terminal} \\ \sum_i n_i & \text{Node Type} = \text{Exclusive OR} \\ \otimes\, n_i & \text{Node Type} = \text{And} \end{cases} \tag{1}$$

Where N_p is the total number of paths at gate p, n_i is the number of elements for each input into the gate, $\sum$ is the summation over all of the inputs, and $\otimes$ is the Cartesian product over all of the elements of the inputs. For example, for the tree shown in Figure 1, the total number of paths is

$$\begin{aligned} N_{AG} &= \left(\sum_{i=1}^{3} n_i^{OG1} \otimes \sum_{i=1}^{4} n_i^{OG2} \otimes \left(\sum_{i=1}^{3} n_i^{OG3} \otimes \sum_{i=1}^{3} n_i^{OG4} \right) \right) \\ &= 3 \otimes 4 \otimes (3 \otimes 3) = 108 \end{aligned} \tag{2}$$

4. DEVELOPING TREES

Software called LED Tools was developed to aid in developing the possibility trees outlined in the previous sections [5]. Using the software, experts intuitively and visually can develop trees as they analyze each detail of the tree. In this section, we consider the development of a possibility tree with

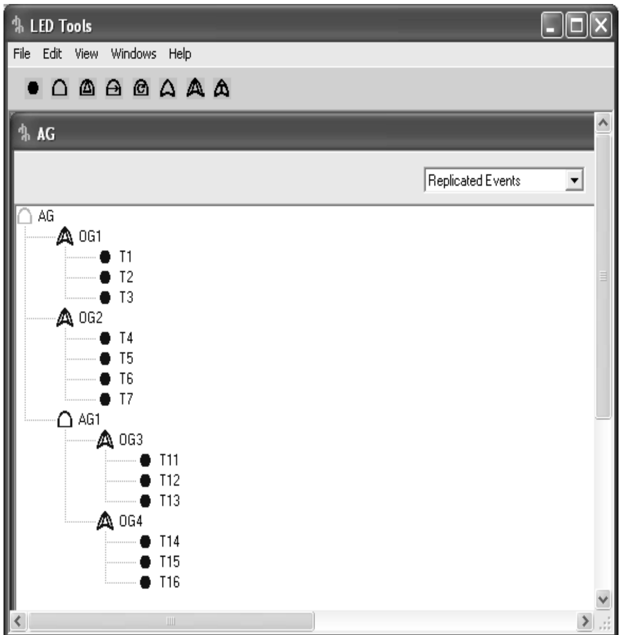

Figure 1: A sample possibility tree. denotes an Exclusive Or gate, denotes an And gate, and denotes a terminal.

the software. One of the applications of possibility trees is in generating risk scenarios for a given system. It is humanly impossible to derive and characterize all possible risk scenarios in a complex system by considering the system as a whole. Usually, experts express more comfort in supplying knowledge of discrete sub-domains, which are within their areas of expertise. A possibility tree supports this idea of domain decomposition by allowing experts to focus on just the few and familiar details of smaller domains without having to be concerned with the overall structure of the model. Figure 2 shows a possibility tree where the scenarios represent the possible ways in which computer information can be lost. This situation is a standard information security issue, and information security analysts usually rely on their expertise to evaluate all of the possible ways in which the end state of information loss can be realized.

In the possibility tree shown in Figure 2, four possible ways are identified (information loss through hacker attacks, virus attacks, power surges,

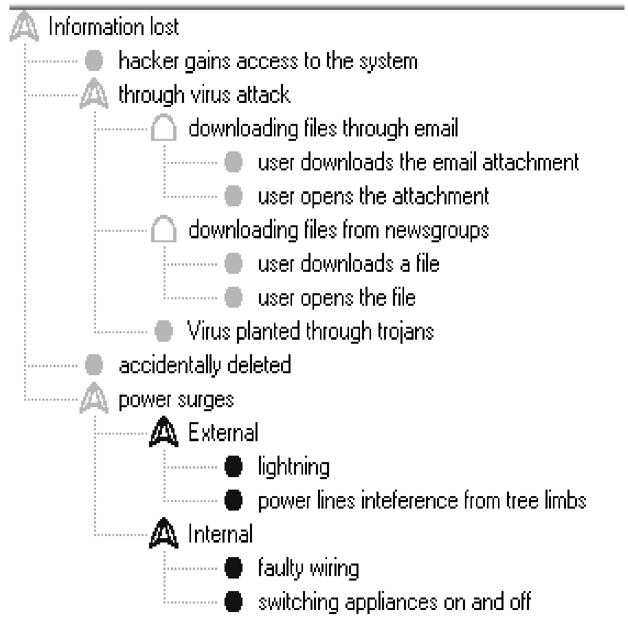

Figure 2: A possibility tree showing the possible ways in which information in computers can be destroyed (excluded nodes are shown in bold).

etc.). Because the example is used for illustration purposes only, only four ways were identified. Additionally, the linguistic descriptions have been shortened to fit the column size allotted for this paper. In reality, experts can identify many more elements and a multitude of scenarios and describe each element in detail. In developing the tree, experts can choose either a top-down or a bottom-up approach for their analysis. A top-down approach is begun with the possible final state; the details then are added to each gate in the tree such that the gates determine the state and the inputs determine how the state can be realized. A bottom-up approach is begun by choosing the initiating event and then adding details to examine the effect of the initiating event. In either approach, multiple experts can be used to work on specific sections of the trees. For the example shown in Figure 3, a top-down approach was found to be more intuitive and the experts are those who have experience in viruses and power surges. The system particulars for the given level of detail are encoded as nodes and branches in the possibility tree and then are combined consequently to form various risk scenarios according to the logic encoded in the gates. Unlike fault trees and event trees, it is not required that the static structure of the system be known before a possibility tree is built and the gates, as shown in Figure 2, encode logic that facilitates aggregation of expert knowledge expressed in natural language. After the details of the first level of the tree are developed, experts then can focus on the new nodes to determine the possible ways in which each of these can be realized. For example, for the tree shown in Figure 3, experts with experience in computer viruses can decide to add more detail to the node—"through virus attack" and add three other nodes—"through email", "newsgroups", and "Trojans" to elaborate on how a virus can infect the system. Therefore, by developing each node in the tree progressively, experts can represent their subjective knowledge about the true behavior of the underlying system behavior, even when the comprehension of the system is qualitative. Collecting such information one node at a time allows analysts to reveal even the least evident characteristics of the system. This collection provides the flexibility to model the details of a highly subjective, dynamic, and sparsely understood system without being burdened by the need to represent the boundaries of the system. The complexity of the model increases significantly as more and more details are added to the possible scenarios, and a possibility tree allows experts to manage this complexity by enabling them to focus on details only at a certain level. By focusing on only a few details at a time, experts can dissect key elements completely and thoroughly before analyzing the entire system.

1: Information lost, hacker gains access to the system
2: Information lost, through virus attack, Virus planted through Trojans
3: Information lost, accidentally deleted
4: Information lost, power surges
5: Information lost, through virus attack, downloading files through email, user downloads the email attachment, user opens the attachment
6: Information lost, through virus attack, downloading files from newsgroups, user downloads a file, user opens the file
6 Paths

Figure 3: Generated paths for the tree shown in Figure 2.

As can be seen in Eqs. (1) and (2), the number of paths for a tree can multiply rapidly as more inputs are added to the tree. In LED Tools, experts have control over the exclusion or inclusion of nodes at the desired level of detail and thus can trim the tree to the appropriate detail in which they are interested. This option allows experts to limit the analysis to certain sections of the possibility tree and thereby enables a more tractable comparative study of the paths. For the tree shown in Figure 2, the nodes that are shown in bold are excluded from the paths. Another benefit that is available in LED Tools and that supports the development of a possibility tree is the availability of structures known as replicants. Replicants are structures in possibility trees (sub-trees or individual nodes) that occur in more than one location. They allow experts to create a structure once and consequently use it in multiple locations that are exactly alike in behavior and structure. Although replicants do not affect the logical structure of the tree, their usage, depending on their multiplicity, results in significant performance gains in the development of the trees. In large trees, SMEs usually discover replicants as they build the trees and identify sections of the tree that use the same structures.

Once the tree is finalized, the paths for the tree can be generated for further analysis. These analyses usually have included risk analysis for the system using each path as a scenario and developing models to determine the effect of risk reduction measures on reducing the risk arising from each scenario [4]. The paths for the tree given in Figure 2 are shown in Figure 3.

In addition to studying the behavior of a system through the paths, an expert, as mentioned in Section 2, also can examine the system through graphical diagrams such as digraphs. Digraphs are a convenient way of examining the paths and the interactions between various elements of a possibility tree. For example, Figure 4 shows the digraph that was generated automatically by LED Tools for the tree shown in Figure 2.

Thus, by allowing experts to develop various component parts of the system progressively and by providing a visual and logical structure for assembling subjective assessments, a possibility tree

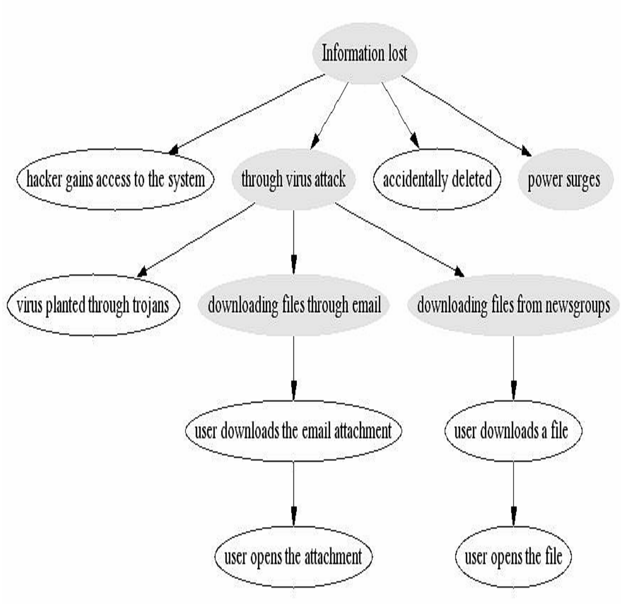

Figure 4: Path digraph generated from the tree shown in Figure 2. The digraph shows the dependencies and the flow of each path.

provides a strong framework for defining even vaguely understood systems.

5. CONCLUSIONS

- Many engineering systems, wherein the systems are only vaguely understood or the threats to the systems are only partially known, are not accessible to analysis using traditional methods such as fault tree analysis.
- A possibility tree models vaguely understood systems by offering an analyst a hierarchical and structured scheme to represent heuristic and subjective expert knowledge in a compact tree form.
- A possibility tree uses two main types of nodes: gates and terminals. There are multiple-gate types but only one terminal type that represents a leaf of the tree and denotes the final event of the tree. The gates represent a decision node from which branching occurs and define the aggregation scheme for each of the branched sub-trees. The aggregation scheme determines the form of combination of various details in the tree.
- Although possibility trees can be developed with significant details, experts can limit the number of paths generated by terminating or excluding the gates at the appropriate level. This process allows experts to focus only on those sections of the tree that are important for the analysis.
- Once the analysis of the paths of a possibility tree is completed, experts can choose a subset of the paths for further analysis, such as risk or decision analysis.

6. REFERENCES

[1] J. R. Thomson, **Engineering Safety Assessment**, John Wiley and Sons, New York, New York, 1987.

[2] S. W. Eisenhawer and T. F. Bott, "A Logic Model for Cook-Off in High Explosives," LA-UR-03-2483, **Proc. 21st International System Safety Conference**, Ottawa, Canada, August 2003.

[3] S. W. Eisenhawer, T. F. Bott, and D V Rao, "Assessing the Risk of Nuclear Terrorism Using Logic Evolved Decision Analysis," LA-UR-03-3467, **Proc. 2003 ANS Annual Meeting**, San Diego, California, June 2003.

[4] S. W. Eisenhawer, T. F. Bott, and R. E. Smith, "An Approximate Reasoning Based Method for Screening High-Level-Waste Tanks for Flammable Gas," **Nuclear Technology**, Vol. 130, 2000.

[5] T. F. Bott, S. W. Eisenhawer, J. Kingson, and B. P. Key, "A New Graphical Tool for Building Logic-Gate Trees," LA-UR-03-0135, **ASME-PVP Annual Meeting**, Cleveland, Ohio, July 20–24, 2003.